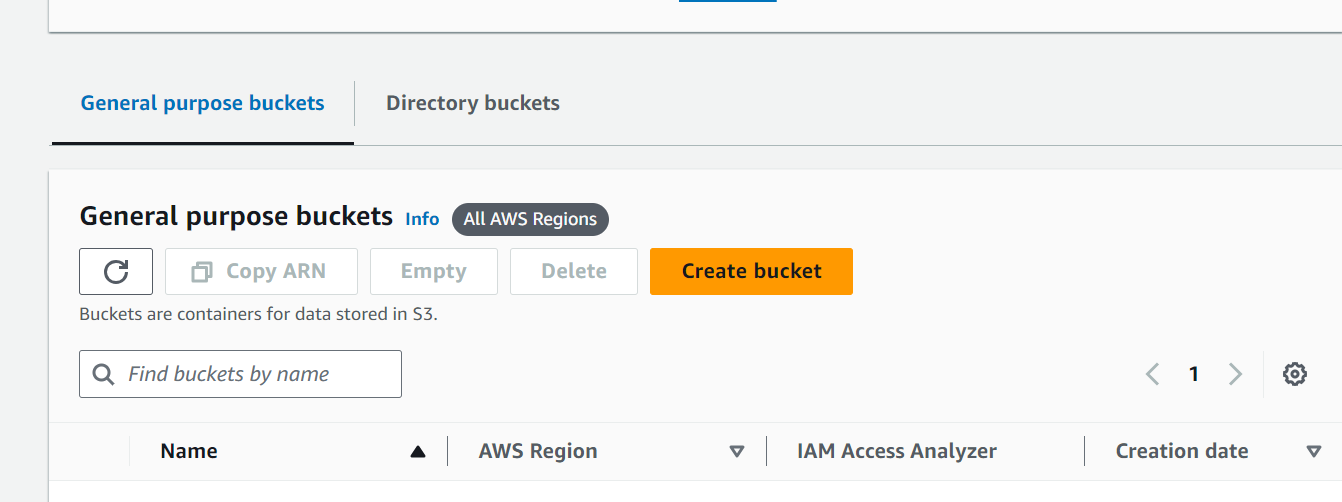
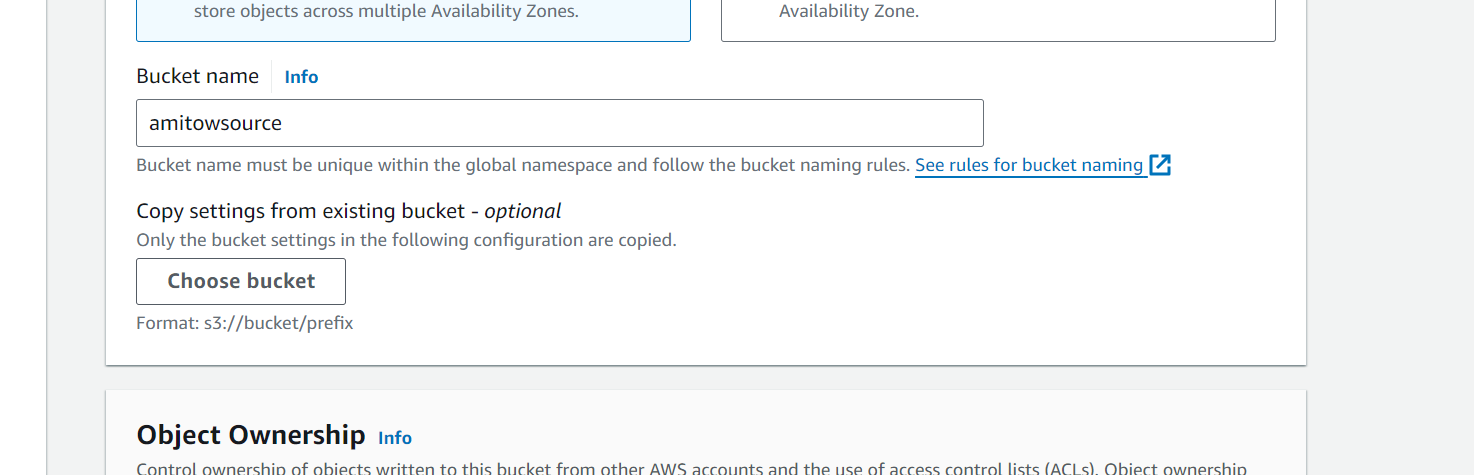
Lambda labs

1. Log in to the AWS Management Console.
2. Create two S3 buckets. One for the source and one for the destination.
3. Create a Lambda function to copy the object from one bucket to another bucket.
4. Test the Lambda Function.

Create Bucket:

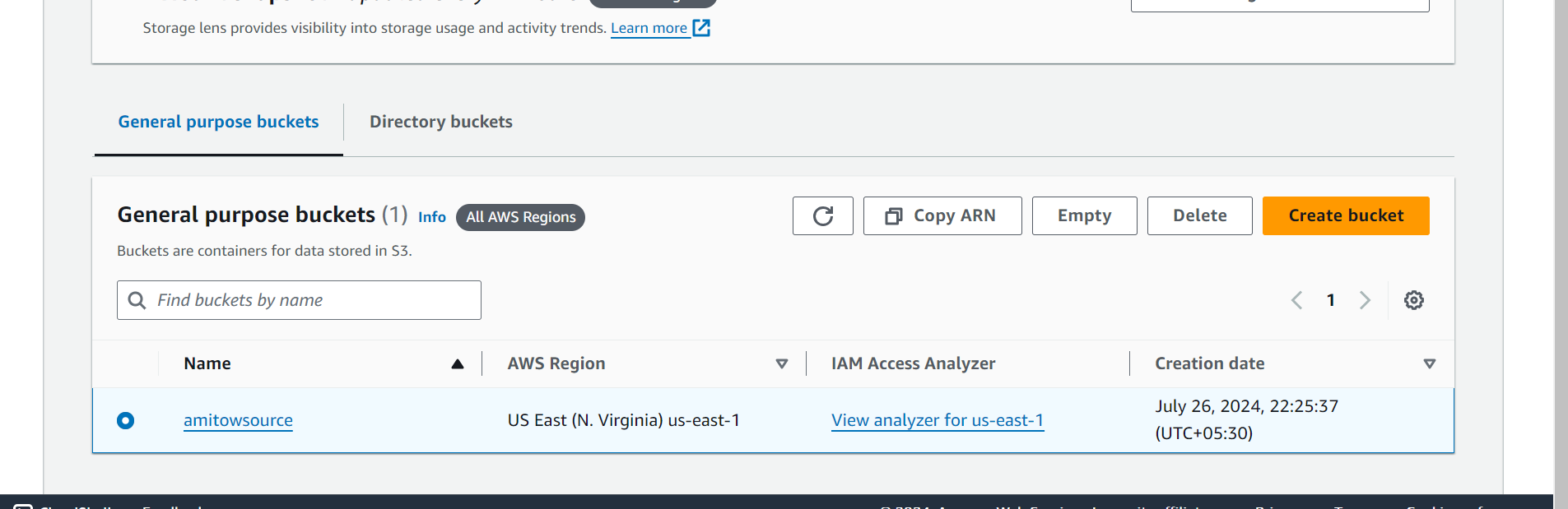




Leave other settings as default and click on the Create button.

Once the bucket is created successfully, select your S3 bucket (click on the checkbox).

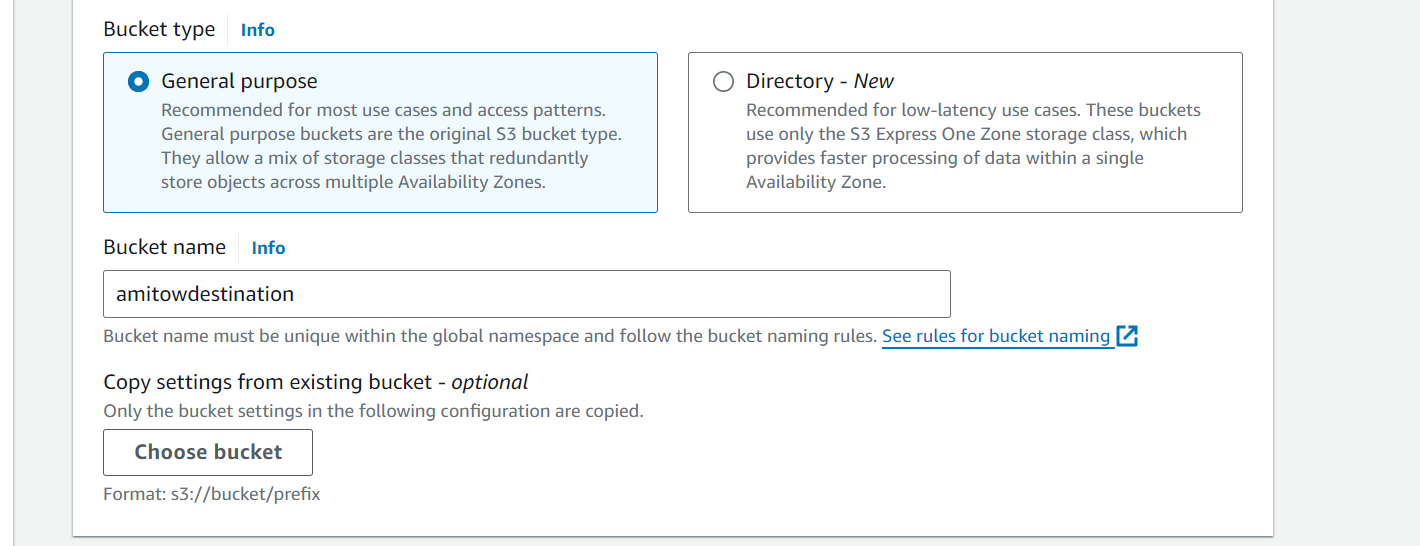
Click on the Copy Bucket ARN to copy the ARN.



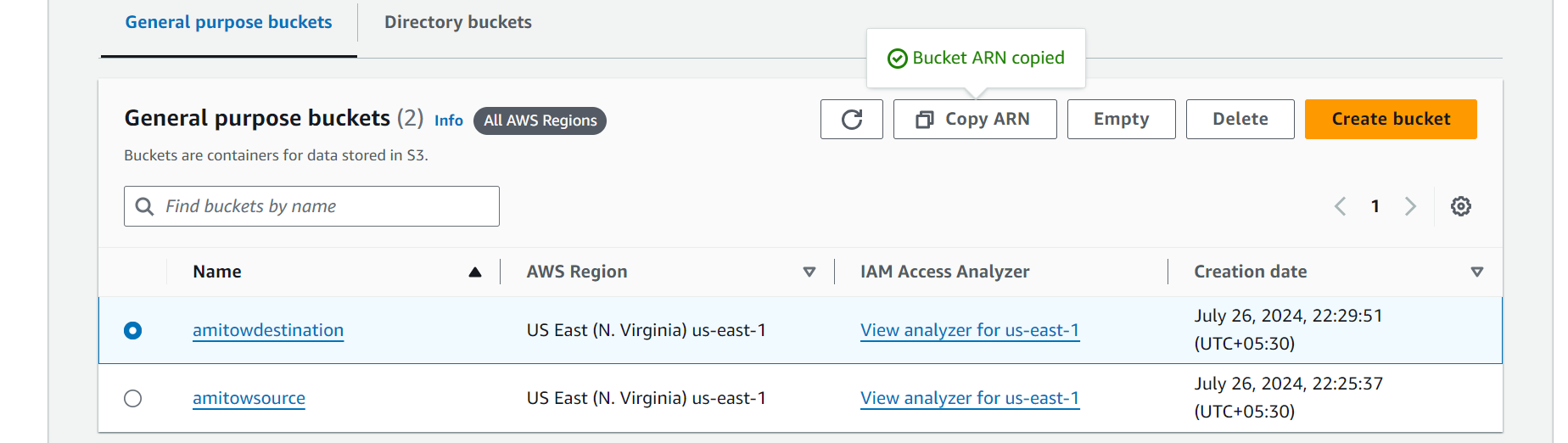
Save the source bucket ARN in a text file for later use.

**Create Amazon S3 Bucket (Destination Bucket)**

Click on Create bucket.



Click on the Copy Bucket ARN to copy the ARN.



Now we have two S3 buckets (Source and Destination). We will make use of our AWS Lambda function to copy the content from source bucket to destination bucket.

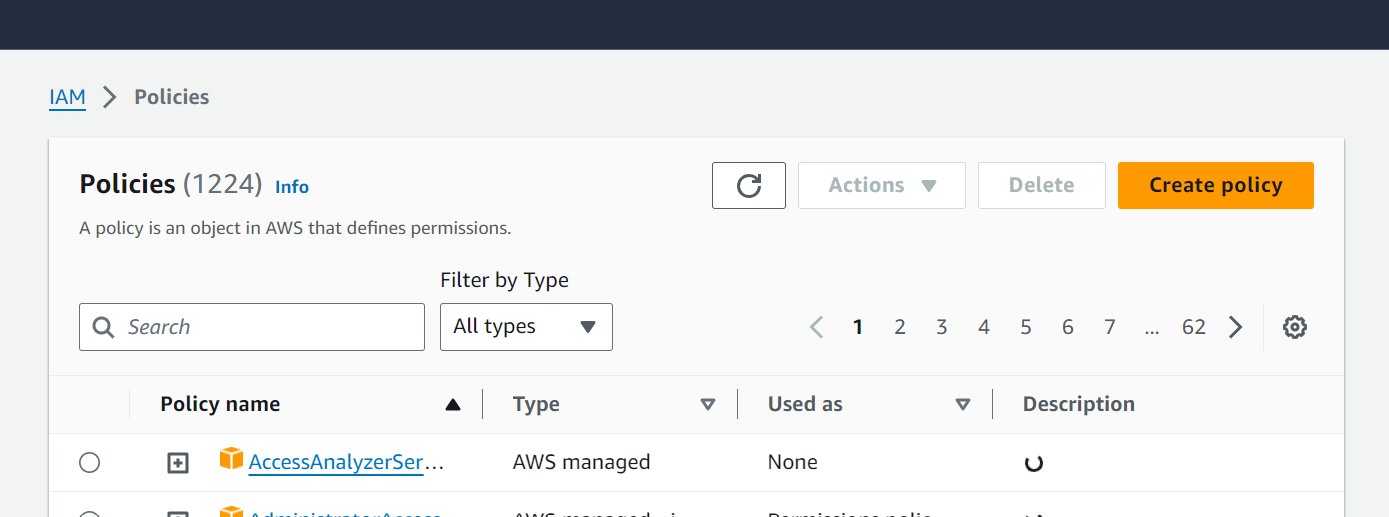
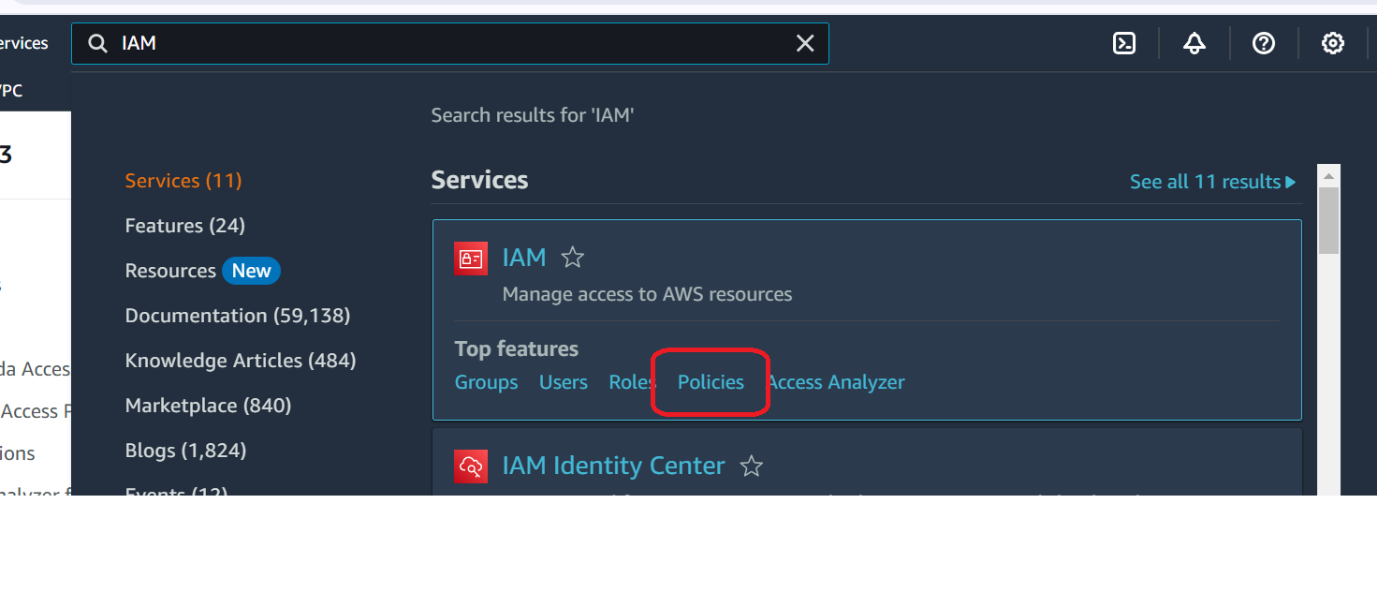
**IAM Configuration**

**Services -> IAM -> Policies**

**Create an IAM Policy**

As a pre-requisite for creating the Lambda function, we need to create a user role with a custom policy.

Click on Create policy.



Click on the JSON tab and copy-paste the below policy statement in the editor:

{

"Version":"2012-10-17",

"Statement":[

{

"Effect":"Allow",

"Action":[

"s3:GetObject"

],

"Resource":[

"arn:aws:s3:::your\_source\_bucket\_name/\*"

]

},

{

"Effect":"Allow",

"Action":[

"s3:PutObject"

],

"Resource":[

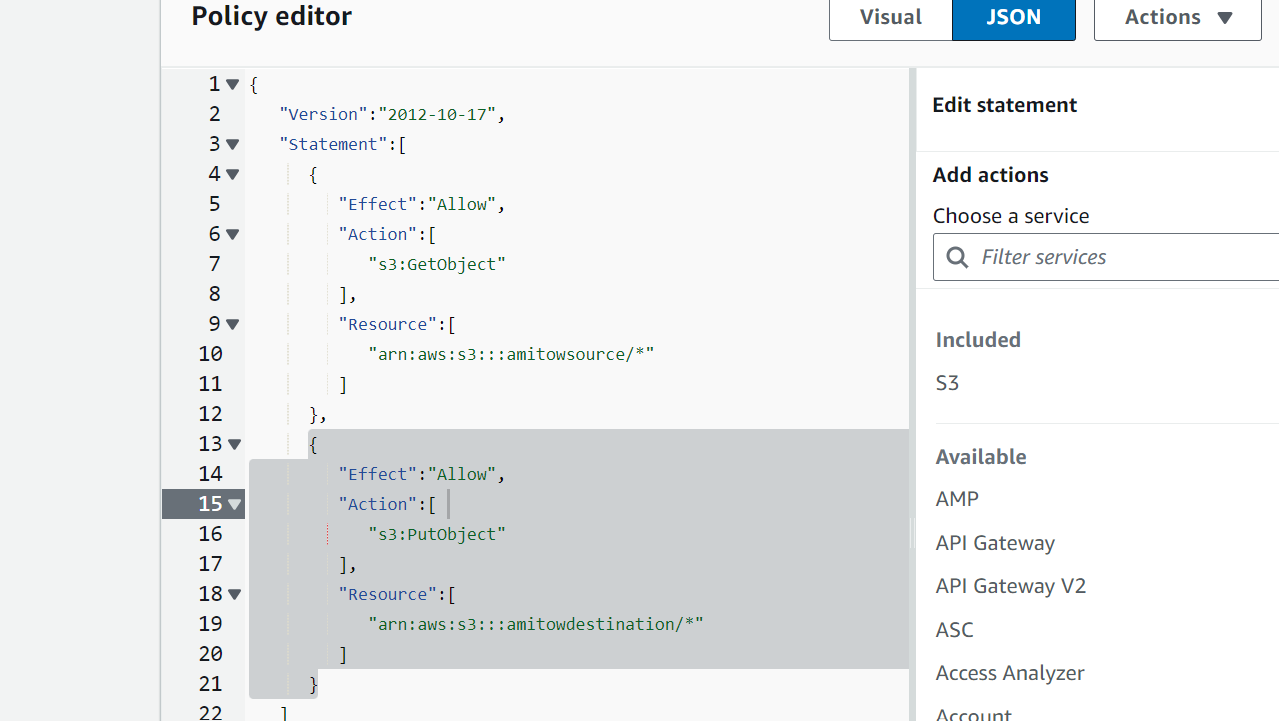
"arn:aws:s3:::your\_destination\_bucket\_name/\*"

]

}

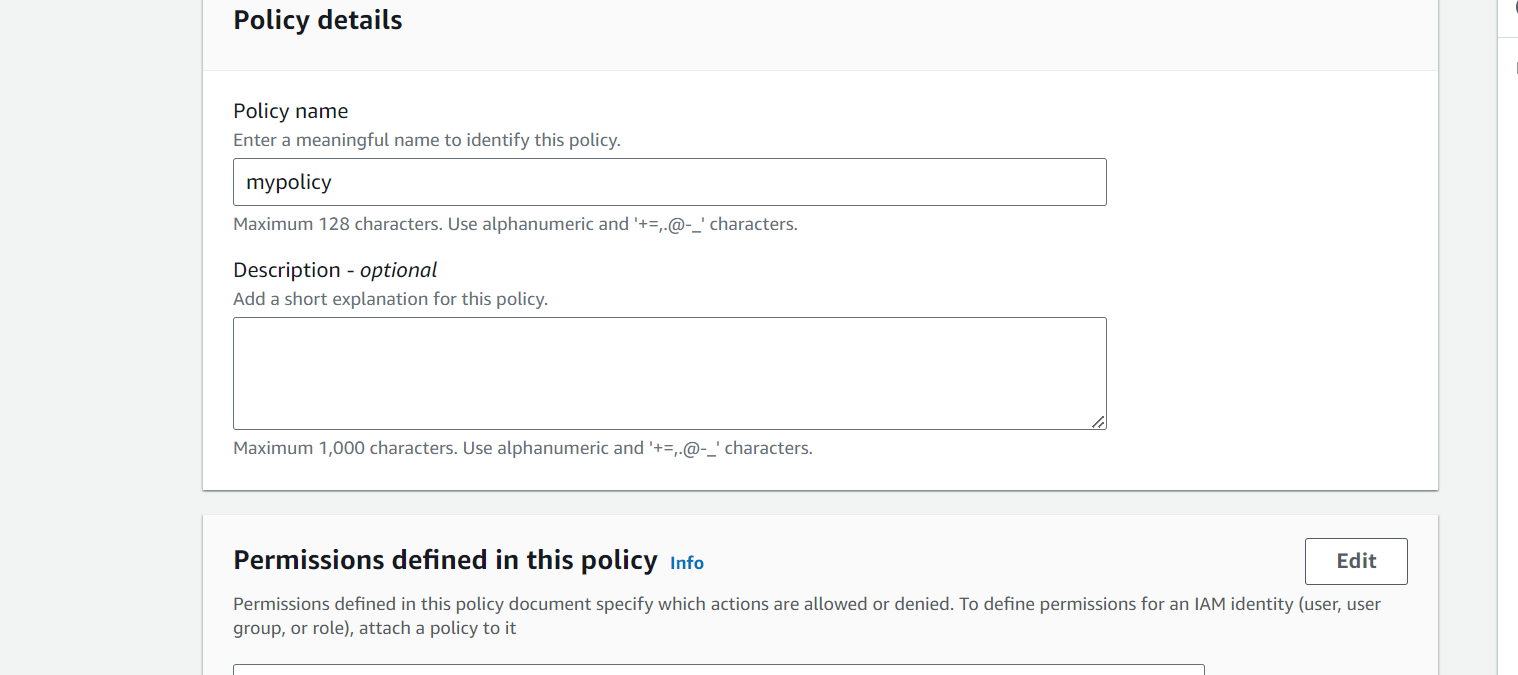
]

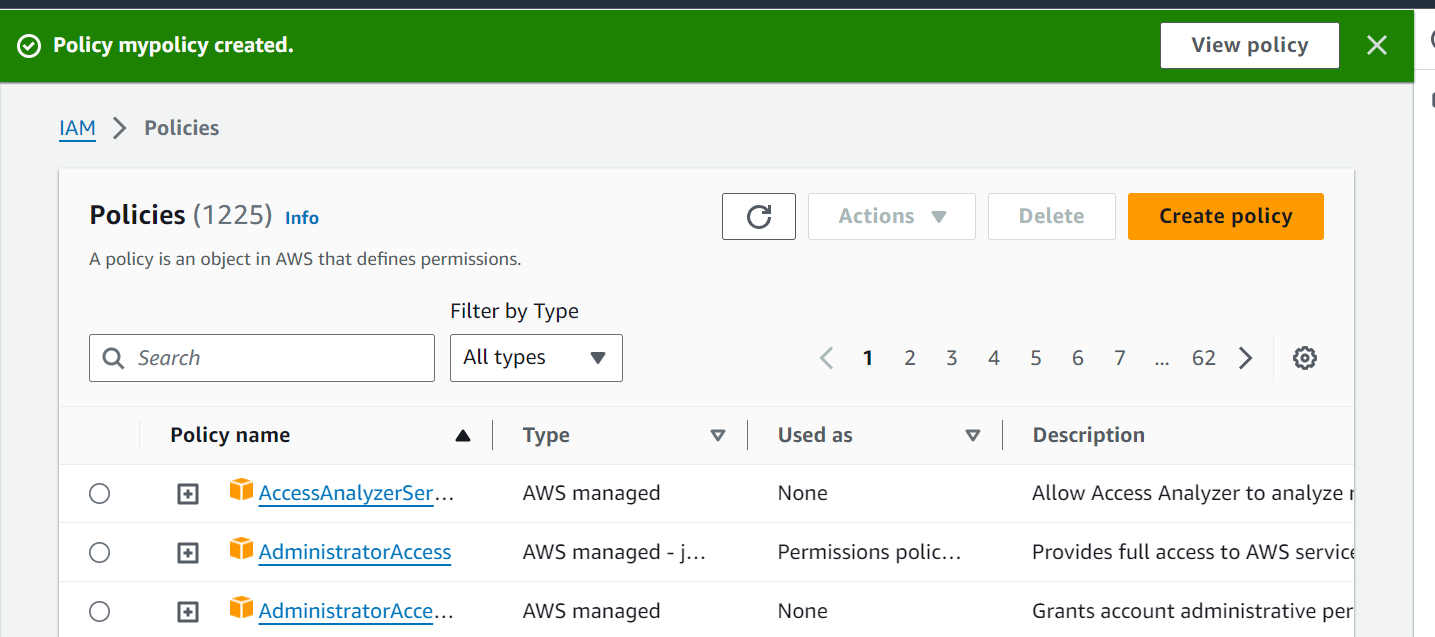
}



On the Create Policy Page:

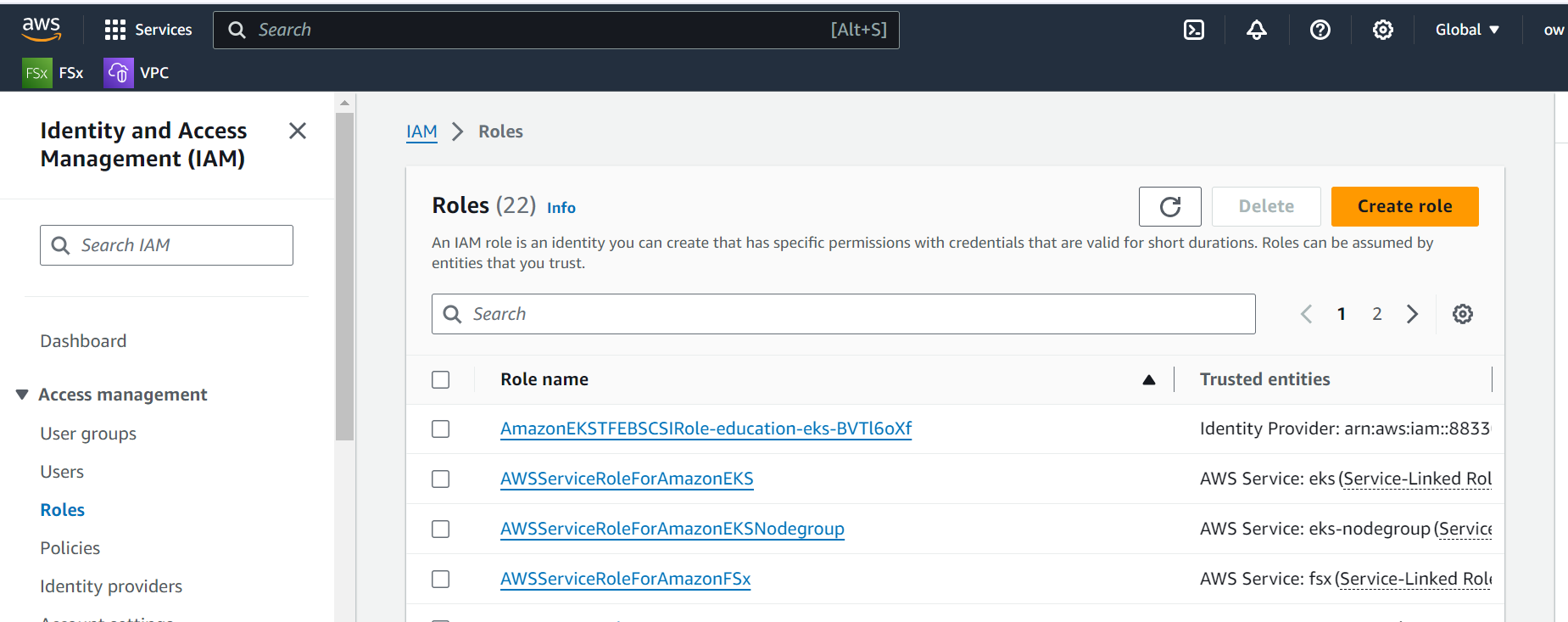
* Policy Name: mypolicy.



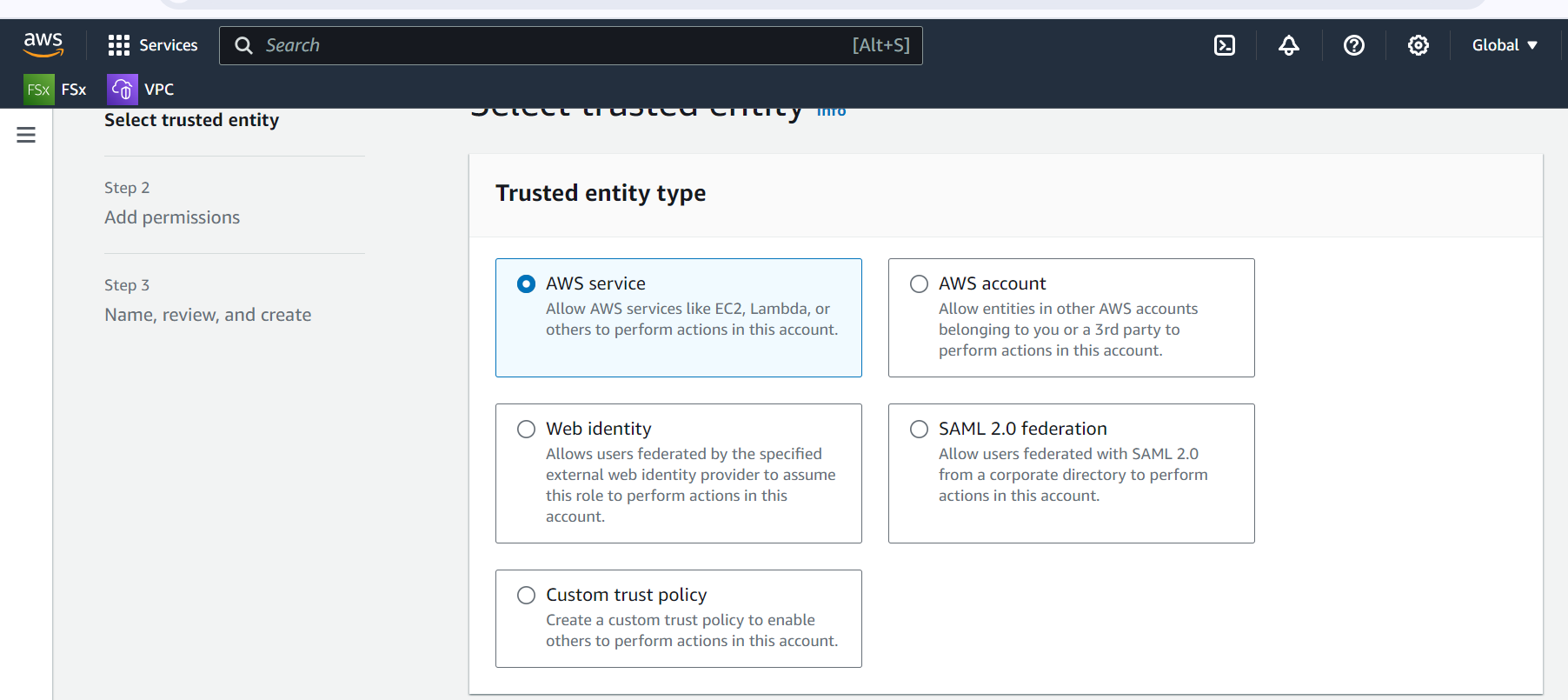


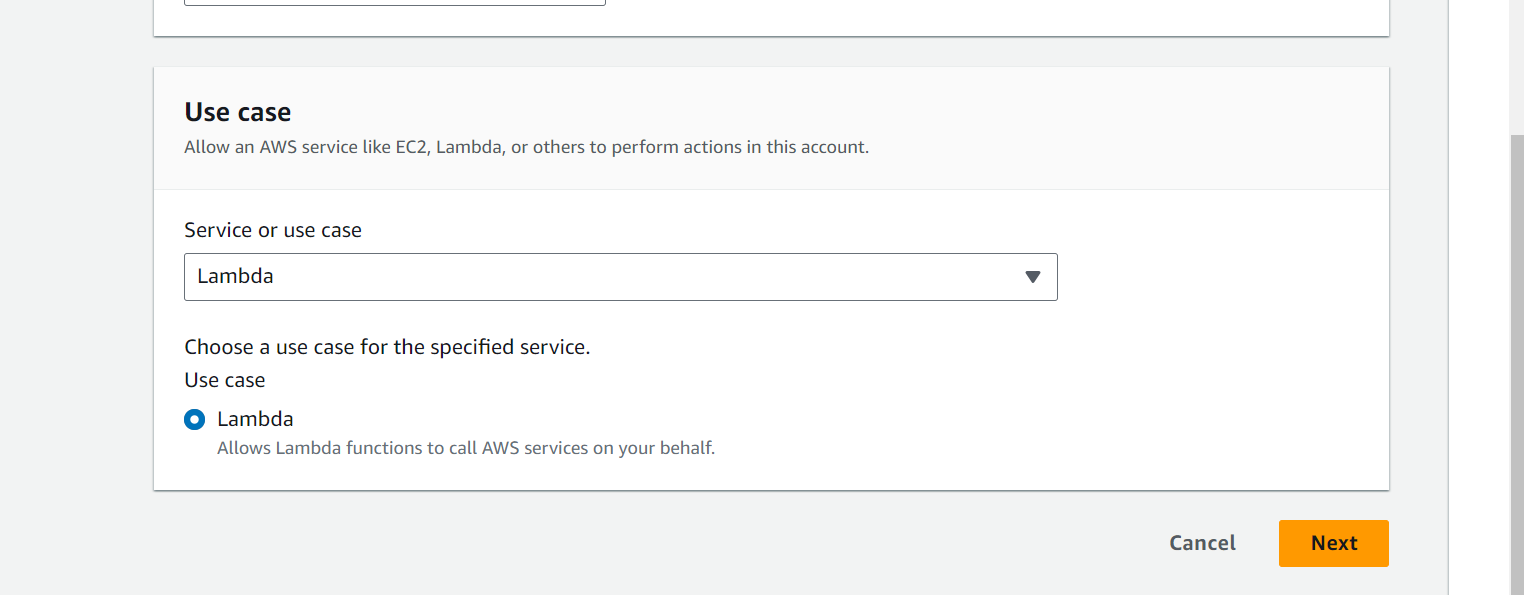
**Create an IAM Role**

In the left menu, click on Roles. Click on the Create role button.

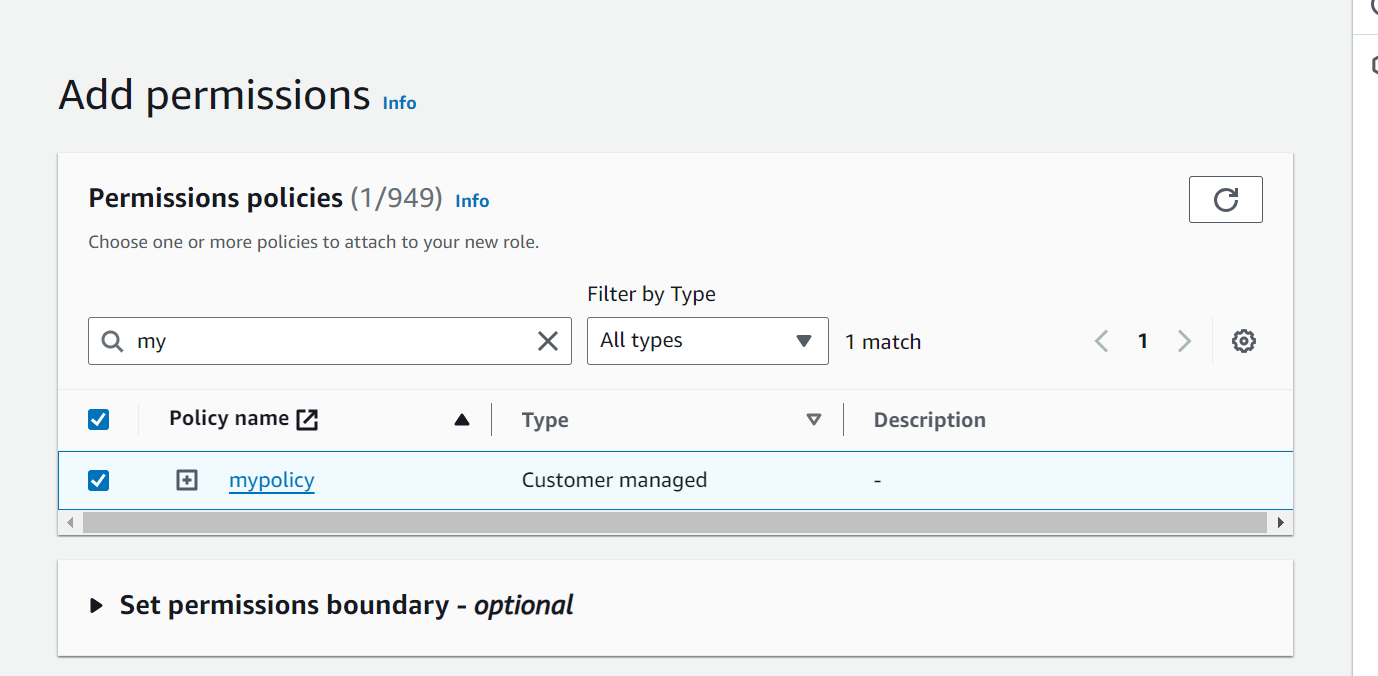


Select Aws service



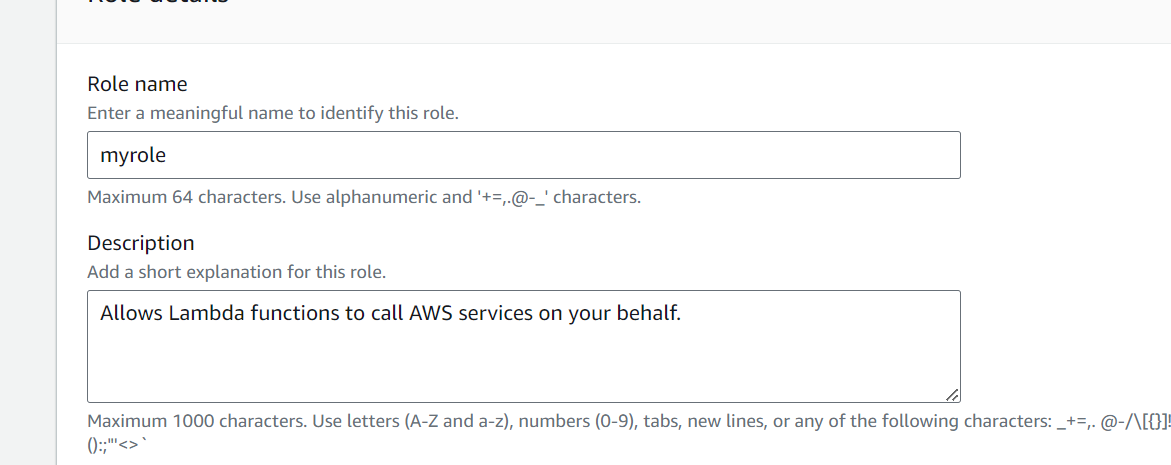


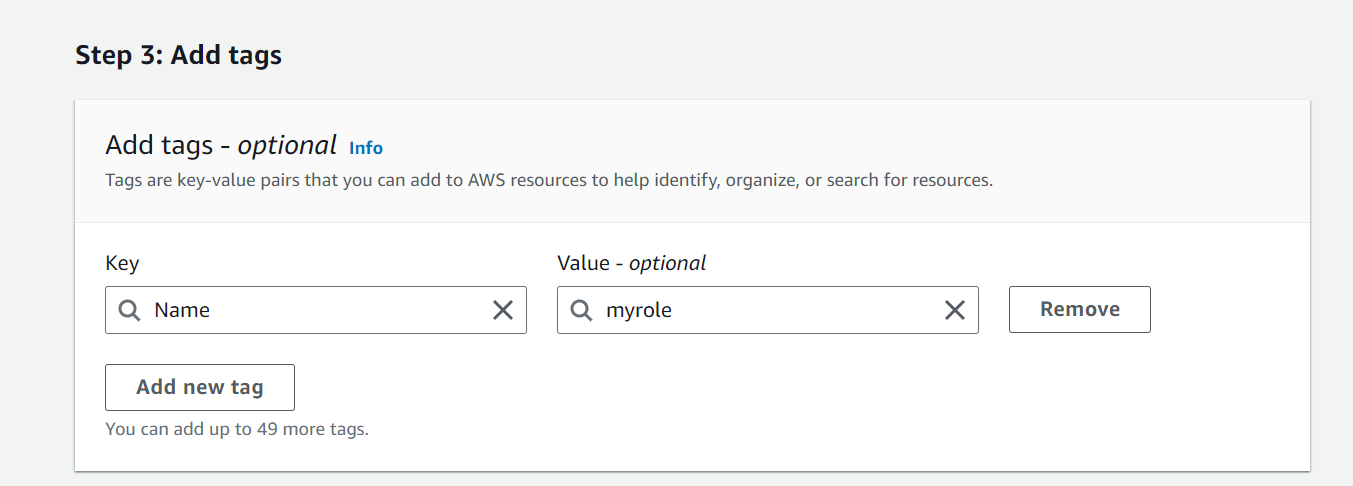
Select policy “mypolicy”



Add Tags: Provide key-value pair for the role:

* Key: Name
* Value: myrole



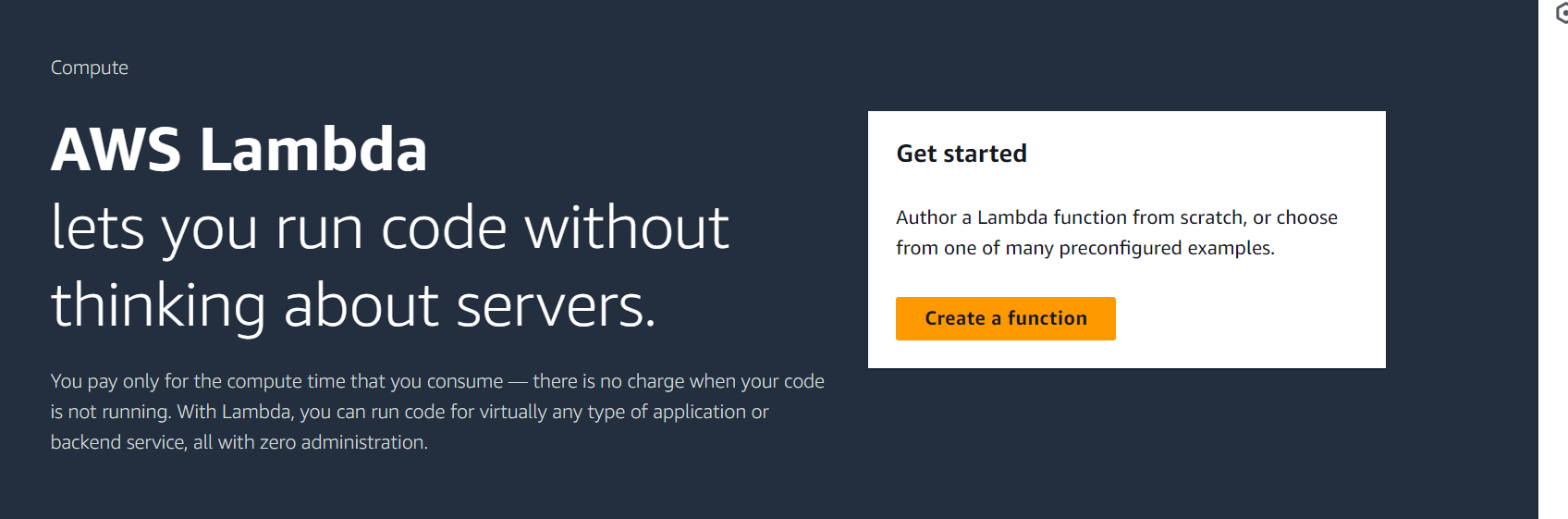
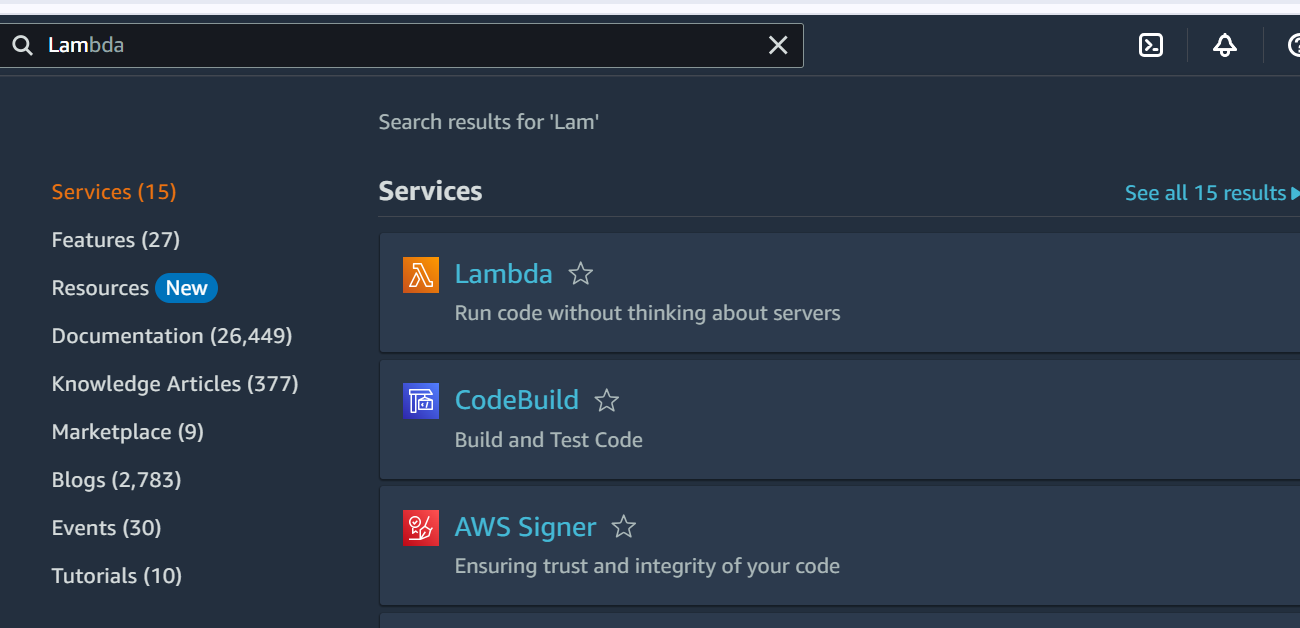


**Lambda Configuration**

**Services -> Lambda**

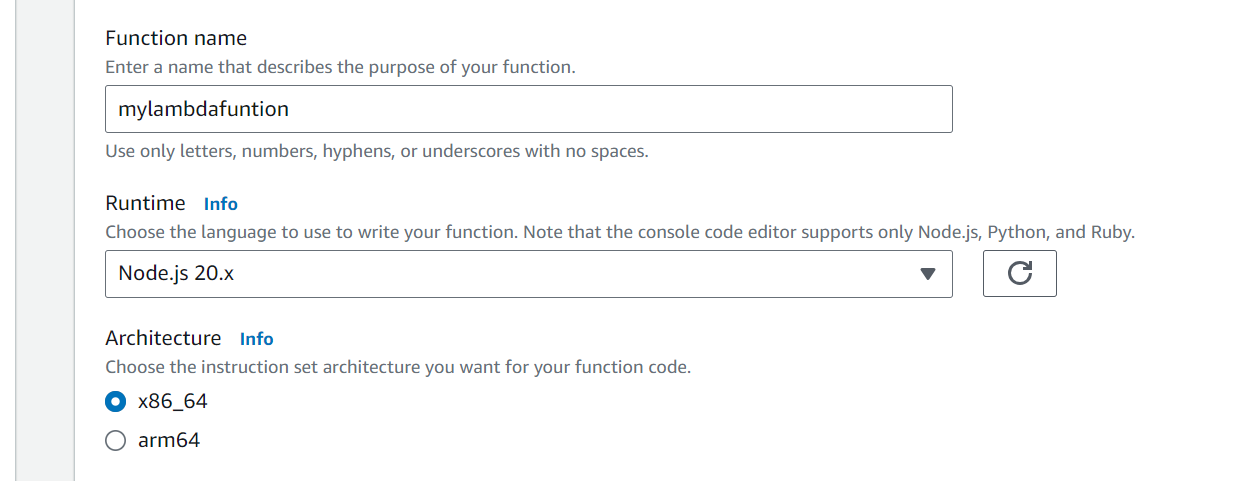
**Create a Lambda Function**

Click on the Create a function button.



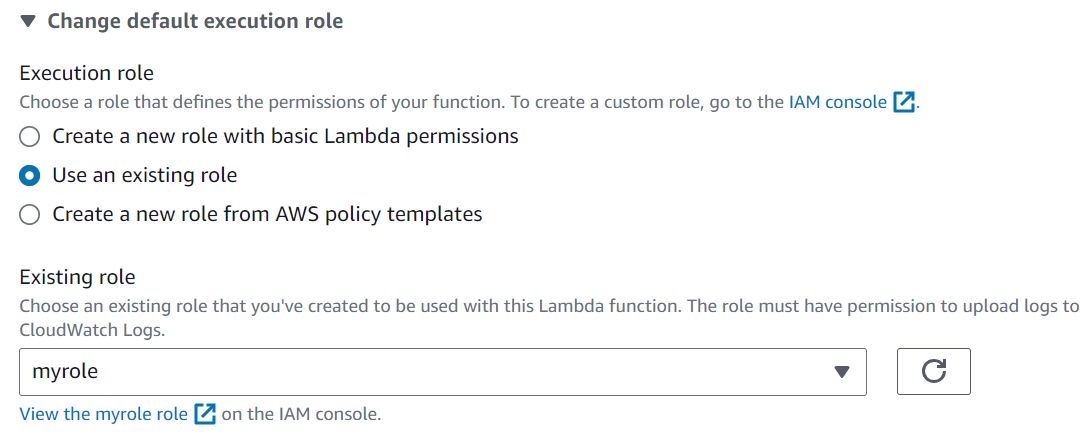
Choose Author from scratch.

* Function name: mylambdafunction
* Runtime: Select Node.js 20x



Role: In the permissions section, select use an existing role.

* Existing role: Select myrole



Click on Create function

From the **Runtime** dropdown list, choose **Python 3.11**.

1. Choose the **Code** tab, and then paste the following Python code:

import boto3

import botocore

import json

import os

import logging

logger = logging.getLogger()

logger.setLevel(logging.INFO)

s3 = boto3.resource('s3')

def lambda\_handler(event, context):

logger.info("New files uploaded to the source bucket.")

key = event['Records'][0]['s3']['object']['key']

source\_bucket = event['Records'][0]['s3']['bucket']['name']

destination\_bucket = os.environ['destination\_bucket']

source = {'Bucket': source\_bucket, 'Key': key}

try:

response = s3.meta.client.copy(source, destination\_bucket, key)

logger.info("File copied to the destination bucket successfully!")

except botocore.exceptions.ClientError as error:

logger.error("There was an error copying the file to the destination bucket")

print('Error Message: {}'.format(error))

except botocore.exceptions.ParamValidationError as error:

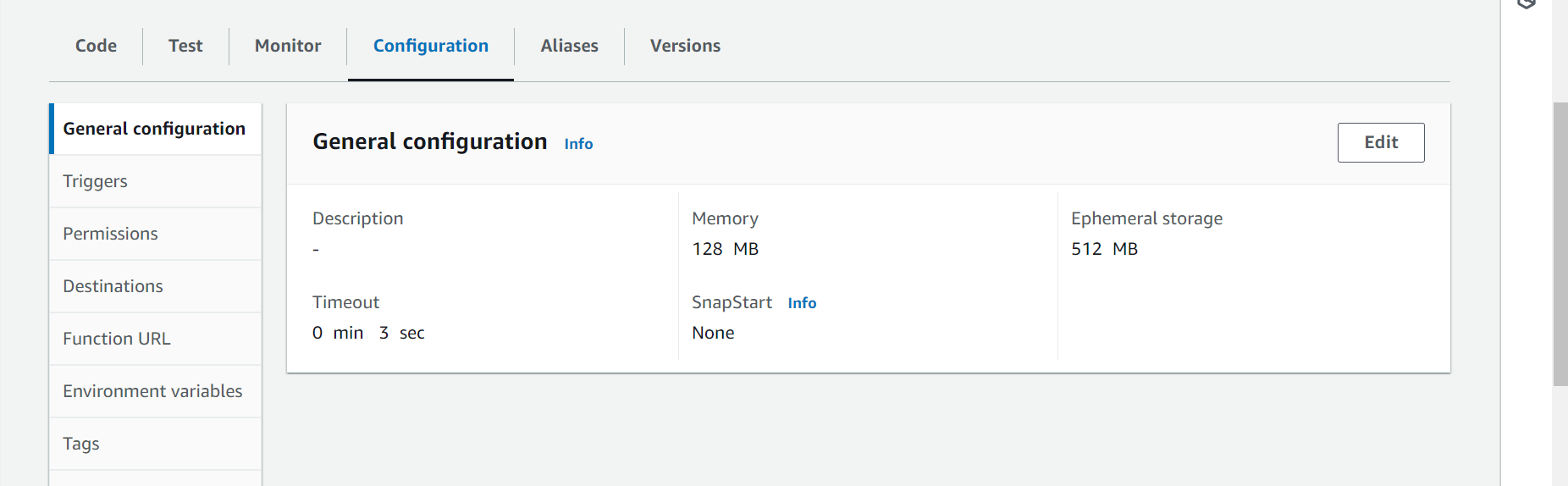
logger.error("Missing required parameters while calling the API.")

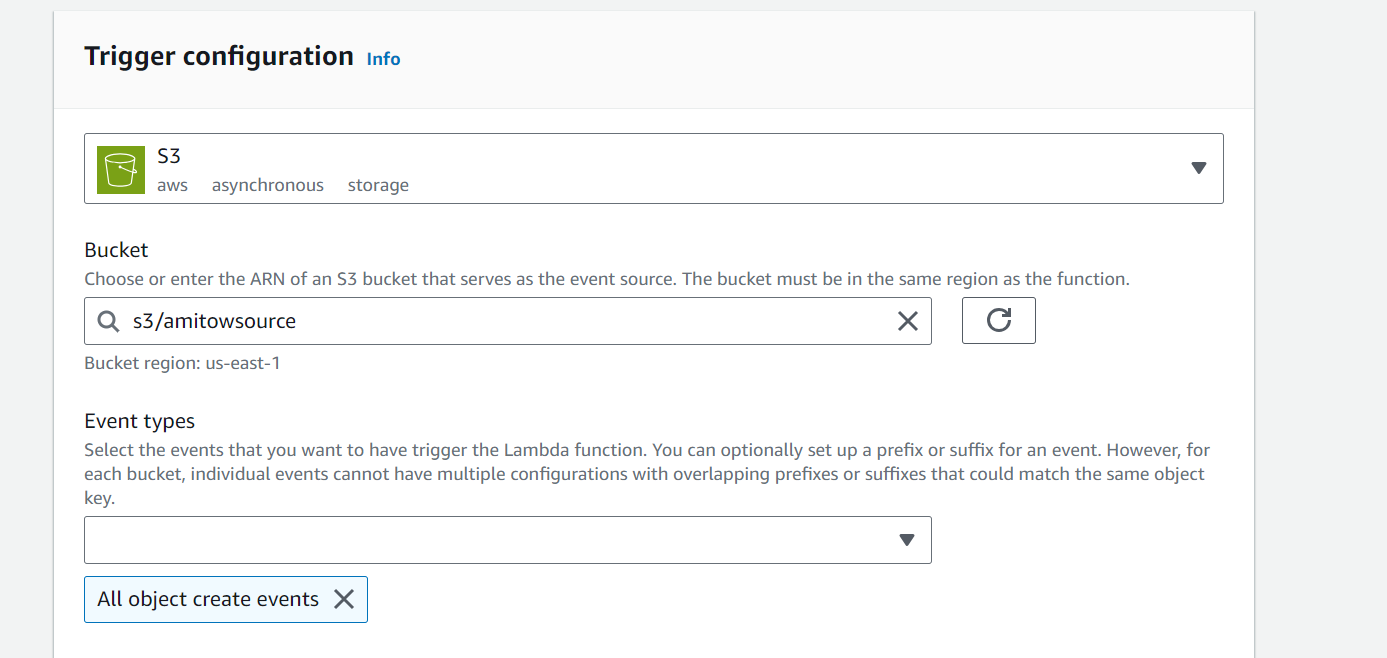
print('Error Message: {}'.format(error))

**Note:** Find the **source\_bucket** name from the event object that the Lambda function receives. You can store the **destination\_bucket** name as an environment variable.

**Adding Triggers to Lambda Function**

Go to the top and left page, click on + Add trigger under **Configuration**`.





Add it

